JOURNAL OF NUMERICAL ANALYSIS AND APPROXIMATION THEORY
J. Numer. Anal. Approx. Theory, vol. 49 (2020) no. 1, pp. 3–14
ictp.acad.ro/jnaat

PRECONDITIONED CONJUGATE GRADIENT METHODS FOR ABSOLUTE VALUE EQUATIONS

NASSIMA ANANE* and MOHAMED ACHACHE†

Abstract. In this paper, we investigate the NP-hard *absolute value equations* (AVE), $Ax - B|x| = b$, where A, B are given symmetric matrices in $\mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$. By reformulating the AVE as an equivalent unconstrained convex quadratic optimization, we prove that the unique solution of the AVE is the unique minimum of the corresponding quadratic optimization. Then across the latter, we adopt the preconditioned conjugate gradient methods to determining an approximate solution of the AVE. The computational results show the efficiency of these approaches in dealing with the AVE.

MSC 2010. 65F08, 90C33, 93C05, 90C20.
Keywords. Absolute value equations, linear system, unconstrained convex quadratic optimization, linear complementarity problems.

1. INTRODUCTION

The *absolute value equations* (AVE) of the type:

$$Ax - B|x| = b, \tag{1}$$

where A and B are given matrices in $\mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $|x|$ denotes the vector with absolute values of each components of the vector x, was investigated by [1], [11], [12], [15]. A special case of (1) when $B = I$ (I denotes the identity matrix) is the AVE of the type:

$$Ax - |x| = b.$$

The AVEs arise in many scientific areas and mathematical problems such as *linear complementarity problems* (LCP), boundary value problems, equilibrium problems and interval linear equations. As is known in [11], the general NP-hard linear complementarity can be formulated as the AVE (1), then it is an NP-hard in its general form. Furthermore, much research has been devoted

* Département de génés des procédures. Faculté de Technologie. Université ferhat Abbas, Sétif 1. Sétif 19000, Algérie, e-mail: nasimaannan@gmail.com.
† Laboratoire de Mathématiques Fondamentales et Numériques. Université ferhat Abbas, Sétif 1. Sétif 19000, Algérie, e-mail: achache_m@univ-setif.dz. This work has been supported by: La Direction Générale de la Recherche Scientifique et du Développement Technologique (DGRSDT-MESRS), under project PRFU number C00L03UN190120190004. Algérie.

to achieve their numerical solutions efficiently (see, *e.g.*, [1], [2], [7], [8], [9], [12], [15]).

In this paper, by reformulating the AVE (1) into an equivalent unconstrained quadratic optimization problem, we prove first under the condition that the smallest singular value of A is greater than the largest singular value of B, the AVE (1) is uniquely solvable for any b. Secondly, we show that the unique minimum of the corresponding unconstrained quadratic problem is the unique solution of the AVE (1). Then across the latter, we apply the conjugate gradient algorithms to approximate numerically the solution of the AVE (1). In the presence of the ill-conditioned, preconditioned conjugate gradient methods can be used to ensure and to accelerate the convergence of the basic CG algorithms. We show across some examples of the AVE, the efficiency of these approaches.

Now we describe our notation. The scalar product of two vectors x and y in $\mathbb{R}^n$ is denoted by x^Ty. For $x \in \mathbb{R}^n$, the norm $\|x\|$ will denote the Euclidean norm $(x^Tx)^{1/2}$, and $\operatorname{sign}(x)$ will denote a vector with components equal to $+1, 0$ or -1, depending on whether the corresponding component of x is positive, zero or negative, respectively. In addition, $D(x) := \partial|x| = \operatorname{sign}(x))$ will denote the diagonal matrix corresponding to $\operatorname{sign}(x)$ where $\partial|x|$ denotes the generalized Jacobian for the absolute value $|x|$ based on a sub-gradient. The vector of ones and the inverse of a nonsingular matrix A are denoted, respectively, by e and A^{-1}. $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ stand for the minimal and the maximal eigenvalues of a matrix M.

As it is known, for a symmetric matrix M, the minimal and the maximal singular values of M, are defined by $\sigma_{\min}(M) = \min_{\|y\|=1} \|My\|$ and $\sigma_{\max}(M) = \|M\| = \max_{\|y\|=1} \|My\|$, respectively. Here, $\|M\|$ is called the spectral induced norm. Finally, the spectral condition number of a nonsingular symmetric matrix M is denoted by $\kappa(M) = \frac{|\lambda_{\max}(M)|}{|\lambda_{\min}(M)|}$.

The paper is built as follows. In Section 2, the unique solvability of the AVE (1) as well as its equivalence reformulation to unconstrained quadratic optimization and the basic conjugate gradient methods for solving the AVE (1) are stated. In Section 3, the preconditioned conjugate gradient algorithms are proposed. Numerical results are reported in Section 4. The paper is ended with a conclusion and future work in Section 5.

2. BASIC CONJUGATE GRADIENT METHODS

Before describing the conjugate gradient algorithm, the following results are useful. For given symmetric matrices A and B, we define, for any diagonal matrix D whose elements are equal to $1, 0$ or -1, the matrix $Q = A - BD$. To prove the unique solvability of the AVE (1), the following result is required.

LEMMA 1. *If symmetric matrices A and B satisfy:*

$$\sigma_{\min}(A) > \sigma_{\max}(B),$$

then the matrix $A - BD$ is nonsingular for any diagonal matrix D whose elements are equal to $+1$, 0 or -1.

Proof. Assume the contrary, that $A-BD$ is singular, then for some nonzero vector x with $\|x\| = 1$, we then have that $(A - BD)x = 0$, which derives a contradiction. This implies that $Ax = BDx$. Hence

$$\begin{aligned}\sigma_{\min}(A) &= \min_{\|y\|=1} \|Ay\| \leq \|Ax\| = \|BDx\| \\ &\leq \|B\|\|D\|\|x\| \leq \|B\| = \sigma_{\max}(B).\end{aligned}$$

This contradicts our condition. Hence $A - BD$ is non-singular. □

Now according to the equality $D(x)x = |x|$, with $D(x) = \operatorname{diag}(\operatorname{sign}(x))$ the AVE (1) can be transformed into the following linear system of equations:

$$Qx = b, \tag{2}$$

where $Q = A - BD$.

Lemma 2. *If symmetric matrices A and B satisfy*

$$\sigma_{\min}(A) > \sigma_{\max}(B),$$

then the AVE (1) is uniquely solvable for any b.

Proof. Based on the result of Lemma 1, the matrix Q is non-singular for any arbitrary diagonal matrix D whose elements are equal to $1, 0$ or -1 and therefore the AVE (1) has a unique solution for any b. □

One of the important numerical tools to solve the system (2) is to transform it into an equivalent quadratic optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \tfrac{1}{2}(Qx - b)^T (Qx - b). \tag{3}$$

The gradient and the Hessian matrix of $f(x)$ are given by:

$$g(x) := \partial f(x) = Q^T (Qx - b)$$

and

$$H(x) := \partial^2 f(x) = Q^T Q.$$

Since $H(x)$ is positive definite for any diagonal matrix D whose elements are equal to $+1$, 0 or -1, the problem (3) has a unique minimum that satisfies

$$g(x) = 0$$

or

$$Q^T Qx = Q^T b. \tag{4}$$

Since Q is non-singular therefore (4) is equivalent to (2) and so is equivalent to AVE (1). Hence solving the AVE (1) is equivalent to find the unique minimum of (3).

The conjugate gradient methods are known to be effective in solving quadratic problems in finite termination [4], [5], [16], [17], [18]. These methods start with an initial point x_0 and generate a sequence $\{x_k\}$ according to the following recurrence formula:

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k = 0, 1, 2, \ldots \tag{5}$$

where $\alpha_k > 0$ is the step-size obtained by a line search and the directions d_k are computed by the rule:

$$d_k = -g_k + \beta_k d_{k-1}, \quad k \geq 1,\ d_0 = -g_0, \tag{6}$$

where β_k is a suitable positive scalar known as the conjugate updating parameter and g_k refers to $g(x_k)$.

2.1. Exact line search. Determining the step-size α_k in (5) along the direction d_k, for an objective function $f(x)$, which is to be minimized, can be simplified to finding the value of $\alpha_k = \alpha$ which consequently minimizes the function:

$$f(x_{k+1}) = f(x_k + \alpha d_k) = m(\alpha).$$

The function $m(\alpha)$ is of a single variable, that is α. Therefore, the α_k is calculated by an exact line search as follows. Using Taylor's expansion, we have,

$$m(\alpha) = f(x_k + \alpha d_k) = f(x_k) + \alpha g_k^T d_k + \tfrac{\alpha^2}{2} d_k^T H_k d_k,$$

so

$$\tfrac{\partial m}{\partial \alpha} = 0 \Leftrightarrow \alpha = -\tfrac{g_k^T d_k}{d_k^T H_k d_k}.$$

Therefore the exact line search is taken as:

$$\alpha_k = -\tfrac{g_k^T d_k}{d_k^T H_k d_k}, \tag{7}$$

where H_k refers to $H(x_k)$.

2.2. Computation of β_k. The coefficients β_k being chosen in such a way that d_k is conjugated with all the preceding directions, in other words

$$d_k^T Q d_{k-1} = 0,$$

then it implies that:

$$d_k^T Q d_{k-1} = -g_k^T Q d_{k-1} + (\beta_k d_{k-1})^T Q d_{k-1} = 0,$$

and so:

$$\beta_k = \tfrac{g_k^T Q d_{k-1}}{d_{k-1}^T Q d_{k-1}}. \tag{8}$$

2.3. Basic conjugate gradient algorithms. We are now ready to state the basic CG algorithms for solving the AVE (1).

- **Step 1.** Choose an arbitrary initial point $x_0 \in \mathbb{R}^n, \epsilon > 0$ and $d_0 = -g_0$, $k = 0$;
- **Step 2.** Compute α_k from (7) and set $x_{k+1} = x_k + \alpha_k d_k$;
- **Step 3.** If $\|Ax_k - B|x_k| - b\| < \epsilon$ then STOP, otherwise compute d_k according to $d_k = -g_k + \beta_k d_{k-1}$ with β_k is computed from (8);
- **Step 4.** Set $k = k + 1$, and go to **Step 2**.

In [4], and [17], it is shown that the convergence of the CG methods is linearly global to the unique minimum x^*. It is known that the convergence of CG methods depends heavily on the condition number $\kappa(Q)$. If $\kappa(Q)$ is close to 1, *i.e.*, if the matrix Q is well-conditioned then CG methods converge fast to the solution. Otherwise, in the presence of ill-conditioned of the matrix Q, these methods have a very slow convergence.

3. PRECONDITIONED CONJUGATE GRADIENT ALGORITHMS

Preconditioning is mainly used in CG methods in order to accelerate their convergence when $\kappa(Q)$ is very far from 1, *i.e.*, when Q is ill-conditioned. Based on this fact, we can consider the preconditioned AVE (1):

$$PAx - PB\,|x| = Pb, \tag{9}$$

where P is a non-singular matrix, called the preconditioner. Obviously, the form (9) is a general form of the AVE (1). For $P = I$, the form (9) reduced to the AVE (1). Again using $D(x)x = |x|$, then (9) becomes the following preconditioned linear system:

$$PQx = Pb. \tag{10}$$

Hence the system (10) has a unique solution if the matrix PQ is invertible. Since P is assumed to be non-singular, then we only prove that Q is non-singular. By Lemma 1, the matrix Q is non-singular for any diagonal matrix D whose elements are $1, 0$, or -1, and consequently, the system (10) has a unique solution and so the preconditioned AVE in (9) is uniquely solvable for each b. Based on this observation, therefore, the equivalent preconditioned quadratic optimization problem is:

$$\min_{x\in\mathbb{R}^n} f_P(x) = \tfrac{1}{2}(PQx - Pb)^T(PQx - Pb). \tag{11}$$

The gradient and the Hessian matrix of f are:

$$g^P(x) := \partial f_P(x) = (PQ)^T(PQx - Pb)$$

and

$$H^P(x) := \partial^2 f_P(x) = (PQ)^T(PQ).$$

It is clear that if $P = I$, the problem (11) reduces to the original problem (3). Also since $(PQ)^T(PQ)$ is positive definite matrix, the problem (11) has a unique minimum that satisfies:

$$g^P(x) = 0,$$

or

$$(PQ)^T(PQx - Pb) = 0,$$

which means that the unique minimum is the unique solution of the preconditioned system and which is in turn the unique solution of the AVE (1). For the preconditioned problem (10), with same manner as the basic CG algorithms, we compute the exact line search α_k and the conjugate parameter β_k along the new preconditioned modified search direction by the formulas:

$$\alpha_k = -\frac{(g_k^P)^T d_k}{d_k^T H_k^P d_k}, \tag{12}$$

and

$$\beta_k = \frac{(g_k^P)^T Q d_{k-1}}{d_{k-1}^T Q d_{k-1}}. \tag{13}$$

Now the preconditioned conjugate gradient (PCG) algorithm for solving the AVE (1) is described as follows.

3.1. Preconditioned conjugate gradient algorithm.

- **Step 1.** Choose an arbitrary $x_0 \in \mathbb{R}^n$, a preconditioner matrix $P, \epsilon > 0$ and $d_0 = -g_0^P$, $k = 0$;
- **Step 2.** Compute α_k from (12) and set $x_{k+1} = x_k + \alpha_k d_k$;
- **Step 3.** If $\|Ax_k - B|x_k| - b\| < \epsilon$ then STOP, otherwise compute d_k according to $d_k = -g_k^P + \beta_k d_{k-1}$ with β_k is computed from (13);
- **Step 4.** Set $k = k + 1$, and go to **Step 2**.

Note that there is no unique strategy for choosing the preconditioning matrix P for the conjugate CG methods. In fact, the strategy of choosing P is based on a such way that the $\kappa(PQ) \ll \kappa(Q)$. For more details see [4].

4. NUMERICAL EXPERIMENTS

In this section, we present some numerical experiments on some examples of solvable AVE (1) to confirm the viability of the PCG algorithms. The experiments are performed with MATLAB 7.9 and carried out on a PC where our tolerance is set to $\epsilon = 10^{-6}$. The initial point and the true solution of AVE (1) are denoted by x_0 and x^*, respectively. Meanwhile, the number of iterations, the elapsed times and the residue are denoted by **Iter**, **CPU** and **RSD**=$\|Ax_k - B|x_k| - b\|$, respectively. In our numerical implementation, the appropriate choice of the preconditioners are $P = \frac{1}{n}I, n \geq 1$, and $P = A^{-1}$.

Example 3. Let the symmetric matrices A, B and the vector b be given as:

$$A = (a_{ij}) = \begin{cases} 4n, & \text{if } i = j, \\ n, & \text{if } |i-j| = 1 \\ 0.5, & \text{otherwise} \end{cases}$$

$$B = (b_{ij}) = \begin{cases} n, & \text{if } i = j, \\ \frac{1}{n}, & \text{if } |i-j| = 1, \\ 0.125, & \text{otherwise} \end{cases}$$

$$b = (548, 647.5, \ldots, 647.5, 548)^T.$$

With the initial points $x_1^0 = (0.001, \ldots, 0.001)^T$ and $x_2^0 = (0.9, \ldots, 0.9)^T$, the computational results with different size of n, are summarized in Table 1.

Size n	x_0		$P = I$ (basic CGA)	$P = \frac{1}{n}I, n > 1$	$P = A^{-1}$
100	x_1^0	Iter	29	21	2
		CPU(s)	0.0186	0.0161	0.0057
		RSD	$6.0870 \cdot 10^{-6}$	$8.4089 \cdot 10^{-6}$	$4.5068 \cdot 10^{-07}$
	x_2^0	Iter	27	19	3
		CPU(s)	0.0178	0.0158	0.0074
		RSD	$6.7790 \cdot 10^{-6}$	$9.1783 \cdot 10^{-6}$	$6.9333 \cdot 10^{-06}$
1000	x_1^0	Iter	33	21	2
		CPU(s)	3.5787	2.4979	0.2273
		RSD	$5.4336 \cdot 10^{-6}$	$8.5594 \cdot 10^{-6}$	$4.5352 \cdot 10^{-08}$
	x_2^0	Iter	31	19	3
		CPU(s)	3.3429	2.0959	0.3293
		RSD	$6.0443 \cdot 10^{-6}$	$9.3010 \cdot 10^{-6}$	$8.8665 \cdot 10^{-15}$
2000	x_1^0	Iter	34	21	2
		CPU(s)	26.6083	16.2365	1.5644
		RSD	$5.7908 \cdot 10^{-6}$	$8.4035 \cdot 10^{-6}$	$2.2684 \cdot 10^{-08}$
	x_2^0	Iter	32	19	3
		CPU(s)	24.4110	14.6529	2.2716
		RSD	$6.4720 \cdot 10^{-6}$	$9.2160 \cdot 10^{-6}$	$1.5029 \cdot 10^{-14}$
3000	x_1^0	Iter	34	21	2
		CPU(s)	90.5026	53.2645	4.9661
		RSD	$8.4709 \cdot 10^{-6}$	$8.3032 \cdot 10^{-6}$	$1.5124 \cdot 10^{-08}$
	x_2^0	Iter	32	19	3
		CPU(s)	83.9471	48.7091	5.7692
		RSD	$9.5473 \cdot 10^{-6}$	$9.1581 \cdot 10^{-6}$	$2.7556 \cdot 10^{-14}$

Table 1. Numerical results for Example 3.

The true solution is $x^* = (\frac{4}{3}, \frac{4}{3}, \ldots, \frac{4}{3}, \frac{4}{3})^T$.

Example 4. The hydrodynamic equations (equilibrium problem [13], is modeled as the following non-differentiable algebraic equations:

$$Bx + \max(0, x) = c,$$

where $B \in \mathbb{R}^{n\times n}, c \in \mathbb{R}^n$ are given. Using the identity

$$\max(a, b) = \tfrac{1}{2}(a + b + |a - b|),$$

equality, the hydrodynamic equation can be reformulated as an AVE (1). We have, $Bx + \frac{1}{2}(x + |x|) = c \Leftrightarrow Ax - |x| - b = 0$ where $A = -(2B + I)$ and $b = -2c$.

Consider now, a randomly hydrodynamic equation where $B \in \mathbb{R}^{n\times n}$ and c are given by:

$$B = (b_{ij}) = \begin{cases} b_{ii} = -25.5, \\ b_{i,i+1} = a_{i+1,i} = -2.5, \\ b_{ij} = 0, \end{cases}$$

and

$$c = (-27, -29.5, \ldots, -29.5, -27)^T.$$

The initial points are $x_1^0 = (0.5, \ldots, 0.5)^T$ and $x_2^0 = (0.9, \ldots, 0.9)^T$. The computational results with different size of n, are summarized in Table 2.

The true solution of this example is $x^* = e$.

Example 5. Given a matrix M and a vector q, the LCP [3], consists in finding w, $z \in \mathbb{R}^n$ such that

$$w \geq 0,\ z \geq 0,\ w - Mz = q,\ z^T w = 0.$$

Letting $w = |x| - x$, $z = |x| + x$, then, $w \geq 0$, $z \geq 0$, and $z^T w = 0$. By substituting w and z in LCP, then an equivalent AVE (1) with $A = (I - M)^{-1}(I + M)$, and $b = -(I - M)^{-1}q$, provided that $(I - M)$ is invertible, is obtained. Note that if x solves the AVE (1), then $z = |x| + x \geq 0$ solves the LCP. Let $M \in \mathbb{R}^{n\times n}$ and q be given as:

$$M = (a_{ij}) = \begin{cases} 0.6, & \text{if } i = j, \\ -0.01, & \text{if } |i - j| = 1, \\ 0, & \text{otherwise}, \end{cases}$$

$$q = -e.$$

The initial points are $x_1^0 = (0.001, \cdots, 0.001)^T$ and $x_2^0 = (0.9, \cdots, 0.9)^T$ and the obtained computational results with different size of n, are stated in Table 3.

The true solution is

$$x^* = (0.8477,\ 0.8618,\ 0.8621, \ldots,\ 0.8621,\ 0.8618,\ 0.8477)^T,$$

and then

$$z^* = (1.6954, 1.7237, 1.7241, \ldots, 1.7241, 1.7237, 1.6954)^T$$

is the solution of LCP.

Size n	x_0		$P=I$ (basic CGA)	$P=\frac{1}{n}I, n>1$	$P=A^{-1}$
100	x_1^0	Iter	11	8	4
		CPU(s)	0.0143	0.0133	0.0081
		RSD	$6.1357\cdot 10^{-6}$	$5.7939\cdot 10^{-6}$	$4.8013\cdot 10^{-8}$
	x_2^0	Iter	10	7	3
		CPU(s)	0.0170	0.0113	0.0068
		RSD	$5.6321\cdot 10^{-6}$	$5.1941\cdot 10^{-6}$	$2.0096\cdot 10^{-6}$
1000	x_1^0	Iter	11	7	4
		CPU(s)	1.1760	0.7838	1.5308
		RSD	$5.8906\cdot 10^{-6}$	$2.5537\cdot 10^{-6}$	$4.8112\cdot 10^{-8}$
	x_2^0	Iter	10	6	3
		CPU(s)	1.1115	0.6666	1.1335
		RSD	$5.4239\cdot 10^{-6}$	$2.2815\cdot 10^{-6}$	$2.0219\cdot 10^{-6}$
2000	x_1^0	Iter	11	6	4
		CPU(s)	8.9451	4.7227	5.5383
		RSD	$5.8609\cdot 10^{-6}$	$5.7002\cdot 10^{-6}$	$4.8116\cdot 10^{-8}$
	x_2^0	Iter	10	5	3
		CPU(s)	8.1684	4.0932	5.3227
		RSD	$5.4023\cdot 10^{-6}$	$5.0782\cdot 10^{-6}$	$2.0226\cdot 10^{-6}$
3000	x_1^0	Iter	11	6	4
		CPU(s)	28.4459	15.5639	13.9749
		RSD	$5.8503\cdot 10^{-6}$	$3.7993\cdot 10^{-6}$	$4.8117\cdot 10^{-8}$
	x_2^0	Iter	10	5	3
		CPU(s)	25.7548	12.8931	10.7083
		RSD	$5.3948\cdot 10^{-6}$	$3.3853\cdot 10^{-6}$	$2.0228\cdot 10^{-6}$

Table 2. Numerical results for Example 4.

Example 6. The matrices A and B are given by:

$$\begin{aligned} A =&(a_{ij}) = \begin{cases} a_{ii} = 10001, & \text{if } i=1, \\ a_{ij} = \frac{1}{i+j-1}, & \text{if } i \neq j, \\ a_{ii} = \frac{1}{i+j-1} + 1, & i=j, i\neq 1, \end{cases} \\ B =&I. \end{aligned}$$

For example, if $n=4$, then:

$$A = \begin{pmatrix} 10001 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{4}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{6}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{8}{7} \end{pmatrix}.$$

The spectrum of A is given by $\{10001, 1.657, 1.0189, 1.0002\}$. Since $\lambda_{\min}(A) = 1.0002 > \lambda_{\max}(I) = 1$, then the AVE is uniquely solvable for any b. The matrix A is ill-conditioned since $\kappa(A) = \frac{10001}{1.0002} = 9999 \gg 1$. Based on $\kappa(A)$, we have deduced that $\kappa(Q) \gg 1$ for some matrices D whose diagonal elements are 1, 0, or -1, which confirms for $n=4$ that Q is ill conditioned. For $n \geq 4$, a careful investigation is needed to confirm the ill-conditioning of Q. In fact,

Size n	x_0		$P = I$ (basic CGA)	$P = \frac{1}{n}I, n > 1$	$P = A^{-1}$
100	x_1^0	Iter CPU(s) RSD	7 0.0152 $2.0378 \cdot 10^{-6}$	5 0.0090 $2.3141 \cdot 10^{-6}$	4 0.0083 $9.3375 \cdot 10^{-6}$
	x_2^0	Iter CPU(s) RSD	6 0.0106 $3.3940 \cdot 10^{-6}$	4 0.0081 $3.8226 \cdot 10^{-6}$	4 0.0104 $7.3068 \cdot 10^{-6}$
1000	x_1^0	Iter CPU(s) RSD	7 1.6359 $1.9182 \cdot 10^{-6}$	4 1.0504 $2.4378 \cdot 10^{-6}$	4 0.9420 $9.34412 \cdot 10^{-6}$
	x_2^0	Iter CPU(s) RSD	6 1.4547 $3.2870 \cdot 10^{-6}$	3 0.6995 $4.1163 \cdot 10^{-6}$	4 0.9617 $7.3117 \cdot 10^{-6}$
2000	x_1^0	Iter CPU(s) RSD	7 7.2802 $1.9091 \cdot 10^{-6}$	4 4.1854 $1.2187 \cdot 10^{-6}$	4 4.1506 $9.3440 \cdot 10^{-6}$
	x_2^0	Iter CPU(s) RSD	6 6.1432 $3.2795 \cdot 10^{-6}$	3 3.0415 $2.0590 \cdot 10^{-6}$	4 4.8948 $7.3116 \cdot 10^{-6}$
3000	x_1^0	Iter CPU(s) RSD	7 20.3593 $1.9059 \cdot 10^{-6}$	3 8.6417 $6.7655 \cdot 10^{-6}$	4 11.5123 $9.3440 \cdot 10^{-6}$
	x_2^0	Iter CPU(s) RSD	6 18.1800 $3.2770 \cdot 10^{-6}$	3 8.6375 $1.3729 \cdot 10^{-6}$	4 11.6423 $7.3116 \cdot 10^{-6}$

Table 3. Numerical results for Example 5.

the matrix A is constructed from the Hilbert matrix, which is known to be ill-conditioned.

Next, for $b = (A - I)e \in \mathbb{R}^n$, and with the initial point $x_0 = (0, 0, \cdots, 0)^T$, the computational results for this example with different size of n, are illustrated in Table 4.

The "*" means that the basic CG and the preconditioned CG with $P = \frac{1}{n}I, n > 1$ algorithms failed.

The true solution of this example is $x^* = e$.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented preconditioned conjugate gradient methods for solving the NP-hard absolute value equations. The obtained numerical results with the preconditioned matrix $P = A^{-1}$ are the best since the number of iterations and the elapsed times are minimum compared with those obtained by the basic conjugate gradient algorithms ($P = I$). We hope that the preconditioned absolute value equations serves as a basis for future research on

Size n		$P = I$ (basic CGA)	$P = \frac{1}{n}I, n > 1$	$P = A^{-1}$
4	Iter	3530	2701	2
	CPU(s)	0.1300	0.0983	0.0050
	RSD	$9.9959 \cdot 10^{-6}$	$9.9952 \cdot 10^{-6}$	$4.1168 \cdot 10^{-16}$
10	Iter	16483	12079	2
	CPU(s)	0.6712	0.4594	0.0051
	RSD	$9.9991 \cdot 10^{-6}$	$9.9983 \cdot 10^{-6}$	$6.753 \cdot 10^{-16}$
1000	Iter			2
	CPU(s)	*	*	0.2494
	RSD			$3.9550 \cdot 10^{-14}$
2000	Iter	*	*	2
	CPU(s)			1.5669
	RSD			$7.2000 \cdot 10^{-14}$

Table 4. Numerical results for Example 6.

other more choice for the preconditioned matrix P to intend an efficient study of the absolute value equations.

REFERENCES

[1] M. Achache, N. Hazzam, *Solving absolute value equations via linear complementarity and interior-point methods*, J. Nonl. Funct. Anal., 2018, pp. 1–10.

[2] M.A. Noor, J. Iqbal, E. Al-Said, *Residual iterative method for solving absolute value equations*, Abstract Appl. Anal., Article ID 406232, (2012), pp. 1–9.

[3] R.W. Cottle, J.S. Pang, R.E. Stone, *The Linear Complementarity Problem*, Academic Press, New-York, 1992.

[4] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, New-York, 1987.

[5] R. Fletcher, C.M. Reeves, *Functions minimization by conjugate gradients*, Computational Journal, **7** (1964), pp. 149–154.

[6] M. Hladick, *Bounds for the solution of absolute value equations*, Comput. Optimiz. Appl., **69** (2018), pp. 243–266.

[7] J. Iqbal, M.A. Noor, K.I. Noor, *On iterative method for solving absolute value equations*, Optimiz. Lett., **6** (2012), pp. 1027–1033.

[8] S. Ketabchi, H. Moosaei, *An efficient method for optimal correcting of absolute value equations by minimal changes in the right hand side*, Comput. Math. Appl., **64** (2012), pp. 1882–1885.

[9] O.L. Mangasarian.*A generalized Newton method for absolute value equations*. Optimiz. Lett. **3**, (2009), pp. 101–108.

[10] O.L. Mangasarian, *A Newton method for linear programming*, J. Optim. Theory Appl., **121** (2004), pp. 1–18.

[11] O.L. Mangasarian, R.R. Meyer, *Absolute value equations*, Linear Algebra Appl., **419** (2006), pp. 359–367.

[12] T. Migot, L. Abdellah, M. Haddou, *Solving absolute value equation using complementarity and smoothing functions*, Comput. Appl. Math., **327** (2018), pp. 196–207.

[13] L. Lehmann, M. Radons, M. Rump, CH. Strom, *Sign controlled solvers for the absolute value equation with an application to support vector machines*, 2010.

[14] J. Rohn, *On unique solvability of the absolute value equations*, Optimiz. Lett., **3** (2009), pp. 603–606.

[15] J. Rohn, *An algorithm for computing all solutions of an absolute value equation*, Optimiz. Lett., **6** (2012), pp. 851–856.

[16] Y. Shi, *Modified quasi-Newton methods for solving systems of linear equations*, Int. J. Contemp. Math. Sci., **15** (2007), pp. 737–744.

[17] G. Zoutendijk, *Nonlinear Programming, Computational Methods*, in: Integer and Nonlinear Programming, J. Abadie, ed., North- Holland, Amsterdam, 1970, pp. 37–86.

[18] N. Ujevic, *A new iterative method for solving linear systems*, Appl. Math. Comput., **179** (2006), pp. 725–730.

Received by the editors: October 11, 2019; accepted: February 27, 2020; published online: August 11, 2020.